МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ   
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕЛЕКТРИЧНОЇ ІНЖЕНЕРІЇ   
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА АВТОМАТИЗАЦІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

НАВЧАЛЬНА ДИСЦИПЛІНА  
Алгоритми та методи обчислень

ЗВІТ

З ПРАКТИЧНИХ РОБІТ

Виконав:

студент групи КН-23-1

Батраков Є.Є.

Кременчук 2024

# Практична робота № 2

**Тема**. Асимптотична складність алгоритмів. Інші нотації

**Мета**: набути практичних навичок у розв’язанні задач на оцінку

асимптотичної складності алгоритмів у Ω, Θ, o, *Ɵ*, *w*-нотаціях.

## Хід роботи

### Завдання:

1. Маємо дві функції *f(n)=4n3-12n+9* і *g(n)* = *n3*. Довести, що *f(n)=Ω(g(n))*

Спрощення

f(n)≥c\*g(n)

4n3−12n+9≥c\*n3

n3(4−c)≥12n−9

Вибір С

4−c>0

4−2=2>0

2n3≥12n−9

Пошук n0

2n2 ≥12− n/9

2n2≥12

n2≥6

Таким чином n0=61/2

1. Задано функції f(n)=n3+2n2-5n+8 і *g(n)* = *n4.* Показати, що f(n)=O(g(n)), використовуючи метод меж.

limn→∞ f(n)/g(n)=(n3+2n2-5n+8)/*n4*

Далі ми розбиваємо це на 4 границі та обраховуємо всі в результаті отримуючи у всіх 0.

Тобто limn→∞ f(n)/g(n)=0

Отже це означає, що функція f(n) є асимптотично обмеженою функцією g(n) і, відповідно, f(n)=O(g(n))

# Контрольні питання

1 Що таке асимптотична складність алгоритму?

Асимптотична складність алгоритму - це оцінка часу або пам'яті, необхідного для виконання алгоритму, яка визначається на основі зростання розміру вхідних даних.

2 Які інші нотації, крім O-нотації, використовуються для вираження асимптотичної складності?

Велика Ω (Омега) нотація

Велика Θ (Тета) нотація

Маленька o (омікрон) нотація

Маленька ω (омега) нотація

3 Як визначити асимптотичну складність алгоритму за допомогою символів Θ і Ω?

Θ-нотація:

* Аналізувати алгоритм для визначення кількості операцій, які він виконує, в залежності від розміру вхідних даних (n).
* Знайти верхню і нижню межі виконання алгоритму (тобто максимально і мінімально можливу кількість операцій, які алгоритм може виконати).
* Якщо ці межі асимптотично "еквівалентні", тобто коли нижня і верхня межі зростання можуть бути виражені однією, то вважається, що алгоритм має асимптотичну складність Θ від цієї функції.

Ω-нотація:

* Проаналізуйте алгоритм для встановлення гарантовано необхідної кількості операцій, дивлячись на те, як змінюється обсяг роботи з збільшенням розміру вхідних даних.
* Визначення функції, що описує цю мінімальну кількість операцій. Ця функція має дати нижню межу для асимптотичної складності алгоритму, тобто вказує на те, що алгоритм не може виконуватися швидше за даний рівень складності, незалежно від обставин.

4 Яка різниця між O-нотацією, Θ-нотацією і Ω-нотацією?

O-нотацією – описує верхню межу асимптотичної складності алгоритму. Це означає, що вона показує максимально можливу кількість операцій (час виконання), які може потребувати алгоритм для виконання.

Ω-нотацією ­– використовується для опису нижньої межі асимптотичної складності алгоритму, тобто вона показує мінімально можливу кількість операцій, які алгоритм вимагатиме

Θ-нотацією – точніше визначає асимптотичну складність, вказуючи на те, що кількість операцій (чи час виконання) алгоритму є асимптотично еквівалентною даній функції з обох сторін (і зверху, і знизу).

5 Які основні властивості інших нотацій, таких як o (маленька о), ω (маленька омега) та 𝑜 (маленька о з верхнім індексом)?

o (маленька о):

Маленька o нотація використовується для вказівки того, що алгоритм зростає асимптотично повільніше, ніж дана функція. Вона описує верхню асимптотичну межу, але в нестрогому сенсі, тобто:

ω (маленька омега):

Маленька ω нотація, навпаки, використовується для вказівки того, що алгоритм зростає асимптотично швидше, ніж дана функція, також в нестрогому сенсі

𝑜 (маленька о з верхнім індексом):

Це те ж саме що й маленька о, але яка може дорівнювати нашій функції.

​